



*National Aeronautics and Space Administration  
Goddard Earth Science  
Data Information and Services Center (GES DISC)*

# README Document for the Synchronous Meteorological Satellite (SMS) 1 Visible Infrared Spin-Scan Radiometer (VISSR) Level 1 Atmospheric and Oceanographic Image Processing System (AOIPS) Data

---

VISSRSMS1L1AOIPS

Last Revised 04/02/2020

Goddard Earth Sciences Data and Information Services Center (GES DISC)  
<https://disc.gsfc.nasa.gov>  
NASA Goddard Space Flight Center  
Code 610.2  
Greenbelt, MD 20771 USA

**Prepared By:**

James E. Johnson

04/02/2020

---

Name  
GES DISC  
GSFC Code 610.2

Date

**Reviewed By:**

Name

mm/dd/yyyy

---

Name  
GSFC Code xxx

Date

Name

mm/dd/yyyy

---

Name  
GSFC Code xxx

Date

Goddard Space Flight Center  
Greenbelt, Maryland

# Revision History

---

<b><i>Revision Date</i></b>	<b><i>Changes</i></b>	<b><i>Author</i></b>
04/02/2020	Original	James E. Johnson
05/21/2020	Various minor fixes	James E. Johnson

# Table of Contents

---

1. Introduction.....	5
1.1 Data Product Description.....	5
1.1.1 The Visible Infrared Spin-Scan Radiometer.....	5
1.1.2 The SMS 1 Satellite.....	6
1.2 Algorithm Background.....	7
1.3 Data Disclaimer.....	7
2. Data Organization.....	8
2.1 File Naming Convention.....	8
2.2 File Format and Structure.....	8
2.3 Key Science Data Fields.....	10
3. Data Contents.....	11
3.1 Data Records.....	11
3.2 Metadata.....	19
4. Reading the Data.....	20
5. Data Services.....	21
5.1 GES DISC Search.....	21
5.2 Documentation.....	21
5.3 Direct Download.....	21
6. More Information.....	22
6.1 Contact Information.....	22
6.2 References.....	22
7. Appendices.....	23
Acknowledgments.....	23
Acronyms.....	23
FORTRAN Code.....	24

# 1. Introduction

---

This document provides basic information on using the Synchronous Meteorological Satellite (SMS) 1 Visible Infrared Spin-Scan Radiometer (VISSR) Level-1 Atmospheric and Oceanographic Image Processing System (AOIPS) product.

## 1.1 Data Product Description

The SMS 1 Visible Infrared Spin-Scan Radiometer (VISSR) Level-1 Atmospheric and Oceanographic Image Processing System data product typically contains a scene of the Earth with radiances that were measured in the visible (0.55 to 0.70 micrometer) and/or IR (10.5 to 12.6 micrometer) wavelengths with a spatial resolution of 0.9 and 8 km, respectively. Files also include time, geolocation, orbit, attitude, and telemetry information. There are three types of data files in this product: one contains IR data, one contains the IR grid information(blank before 1974/10/29), and another contains VIS data. Each data file is structured with an AOIPS label, followed by an IPD label, and then an optional 8 telemetry records followed by a set of data records. Visible data are typically 3904 pixels by either 4000 or 2000 scan lines (5 or 2.5 minute scenes respectively). IR data are typically 976 pixels by either 500 or 250 scan lines (5 or 2.5 minute scenes respectively). A full scan of the Earth was made every 20 minutes. The data were used to make 70mm film negatives and 9.5" positive prints on a Dicomed Image Recording System.

Data for this product are available from 1974/07/01 to 1974/09/19 (with gaps); 1974/10/29; 1974/11/13; from 1975/01/08 to 1975/02/26 (with gaps); from 1975/07/09 to 1975/09/12 (with gaps); 1976/08/26 (one VIS image); and from 1979/02/17 to 1979/04/19 (with gaps). The SMS-1 satellite was initially parked over the equator at longitude 45W on June 7, 1974 viewing the hemisphere below the satellite in support of the Global Atmospheric Research Project (GARP) Atlantic Tropical Experiment (GATE). It was moved to its operational position at 75W on Nov 15, 1974 where it remained until GOES-1 was launched, after which SMS 1 was moved to 105W and placed in stand-by-mode as a backup to GOES-1 or SMS-2. The VISSR experiment was operated by the NOAA National Environmental Satellite Data and Information Service (NESDIS), as well as scientists from NASA Goddard Space Flight Center.

This product was previously available from the NSSDC with the identifier ESAD-00018 (old ID 74-033A-01D).

### 1.1.1 The Visible Infrared Spin-Scan Radiometer

The Visible Infrared Spin-Scan Radiometer (VISSR) flown on SMS 1 provided day and night observations of cloudcover and earth/cloud radiance temperature measurements from a synchronous, spin-stabilized, geostationary satellite for use in operational weather analysis and

forecasting. The two-channel instrument was able to take both full and partial pictures of the earth's disk. The infrared channel (10.5 to 12.6 micrometers) and the visible channel (0.55 to 0.70 micrometer) used a common optics system. Incoming radiation was received by an elliptically shaped scan mirror and collected by a Ritchey-Chretien optical system. The scan mirror was set at a nominal angle of 45 deg to the VISSR optical axis, which was aligned parallel to the spin axis of the spacecraft. The spinning motion of the spacecraft (approximately 100 rpm) provided a west-to-east scan motion when the spin axis of the spacecraft was oriented parallel to the earth's axis. The latitudinal scan was accomplished by sequentially tilting the scanning mirror north to south at the completion of each spin. A full picture took 18.2 min to complete and about 2 min to retrace. During each scan, the field of view on the earth was swept by a linear array of eight visible-spectrum detectors, each with a ground resolution of 0.9 km at zero nadir angle. The infrared portion of the detector measured radiance temperatures between 180 and 315 deg K with a proposed sensitivity between 0.4 and 1.4 deg K.

### 1.1.2 The SMS 1 Satellite

The SMS 1 satellite was successfully launched on May 17, 1974. SMS 1 was the first in a series of geosynchronous meteorological satellites developed by NASA and operated by NOAA, which would evolve into NOAA's GOES system. The spacecraft included three instruments: (1) a visible infrared spin-scan radiometer (VISSR) which provided high-quality day/night cloudcover data and made radiance temperatures of the earth/atmosphere system, (2) a meteorological data collection and transmission system (DCS) which relayed processed data from central weather facilities to small automatic picture transmission (APT) equipped regional stations and collected and retransmitted data from remotely located earth-based platforms, and (3) a space environmental monitor (SEM) which measured proton, electron, and solar X-ray fluxes and magnetic fields.

The orbit of the satellite can be characterized by the following:

Parking Station	Longitude	Date
Pre-Operational	45° W	06/07/1974
Operational	75° W	11/15/1974
Post-Operational	105° W	02/16/1976

Geostationary circular orbit at 35,000 km

Inclination of 1.9 degrees

## **1.2 Algorithm Background**

The SMS 1 VISSR AOIPS data were generated from the spacecraft telemetry, attitude and orbital data. The data were originally processed on IBM 360 computers, and copied to 6250 tapes for archival. Further information on the VISSR instrument and data processing can be found in the “VISSR Data Processing Plan for SMS/GOES Satellites” and the “The GOES/SMS User's Guide”.

## **1.3 Data Disclaimer**

The data should be used with care and one should first read both the “VISSR Data Processing Plan for SMS/GOES Satellites” and the “GOES/SMS 1 User's Guide”. Users should cite this data product in their research.

## 2. Data Organization

---

Data from the SMS 1 Visible Infrared Spin-Scan Radiometer (VISSR) AOIPS product are available from July 1, 1974 to April 19, 1979 (with gaps). Each file typically contains a few minutes of data on the order of up to 4000 visible scans and up to 500 IR scans.

### 2.1 File Naming Convention

The data product files are named according to the following convention:

<Platform>-<Instrument>\_<Level>-<Product>-<Type>\_<DateTime>\_<TapeNumber>-<TapeFile>.<Suffix>

where:

- o Platform = name of the platform or satellite (SMS1)
- o Instrument = name of the instrument and product (VISSR)
- o Level = process level (L1)
- o Product = product name (AOIPS)
- o Type = type of data (VIS, IR, IRG)
- o Date = Data start date and time in UTC in format <YYYY>m<MMDD>t<hhmmss> where
  - 1. YYYY = 4 digit year (1978 - 1979)
  - 2. MM = 2 digit month (01-12)
  - 3. DD = 2 digit day of month (01-31)
  - 4. hh = 2 digit hour of day (00-23)
  - 5. mm = 2 digit minute (00-59)
  - 6. ss = 2 digit second (00-59)
- o TapeNumber = 5 digit number of tape (preceded by 'DD' - primary or 'DC' - backup)
- o TapeFile = 1 digit number of file on the tape
- o Suffix = the file format (always TAP, indicating tape binary data)

File name example: SMS1-VISSR\_L1-AOIPS-VIS\_1974m0701t150500\_DD51981-2.TAP

### 2.2 File Format and Structure

The data are stored as they were originally written in IBM binary (big-endian) record oriented structured files. The files were written on 7-track and 9-track tapes using a blocked FORTRAN format. The first file on the tape is the tape header file with two records containing text encoded information about the tape. This is followed by up to three data files. Each data file on the tape contains an AOIPS label record of size 512 bytes, followed by an IPD label record of size 1244 (IR/IRG) or 4172 (VIS) bytes, followed by either 0, 4 or 8 telemetry records also of size 1244 (IR/IRG) or 4172 (VIS) byte, and then a series of data records also of size 1244 (IR/IRG) or 4172 (VIS) bytes. Two IR/IRG files (SMS1-VISSR\_L1-AOIPS-IR\_1975m0217t190123\_DD33364-1.TAP and SMS1-VISSR\_L1-AOIPS-IRG\_1975m0217t190123\_DD33364-2.TAP) are full global images using 4092 byte records instead of 1244 bytes. Each data file typically contains about 5 minutes or less of data, up to 18 minutes for a

global image. For the contents and layout of the data, see Appendix B of the “VISSR Data Processing Plan for SMS/GOES Satellites”.

During data recovery a total of 2480 data files were extracted from 959 tapes (one tape incorrectly assigned to SMS2). Of these, eight files were exact duplicates of other files, one file was nearly identical except for one bad scan line, one file from 1976 belonged to the GOES-1 collection, and one file contained no data records, just a header, and are not included in the SMS-1 VISSR AOIPS collection. In the end 2469 data files were archived for this collection. All files came from tapes designated by a prefix ‘DD’ (primary) and a five digit number, except for 6 files which came from tapes with a ‘DC’ (backup) prefix.

Caution should be taken as some data records contain corrupted records (such as bad bits or bad record sizes). Of the files, 2439 were from SSEC tapes, and only the following 19 came from Redstone tapes (DD51982, DD52930, DD51983, DD52932, DD69593, DD69599, DD58022):

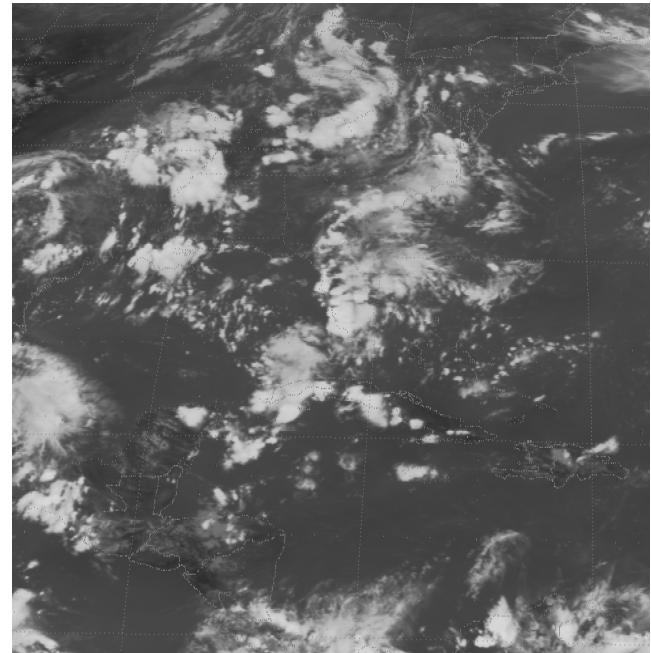
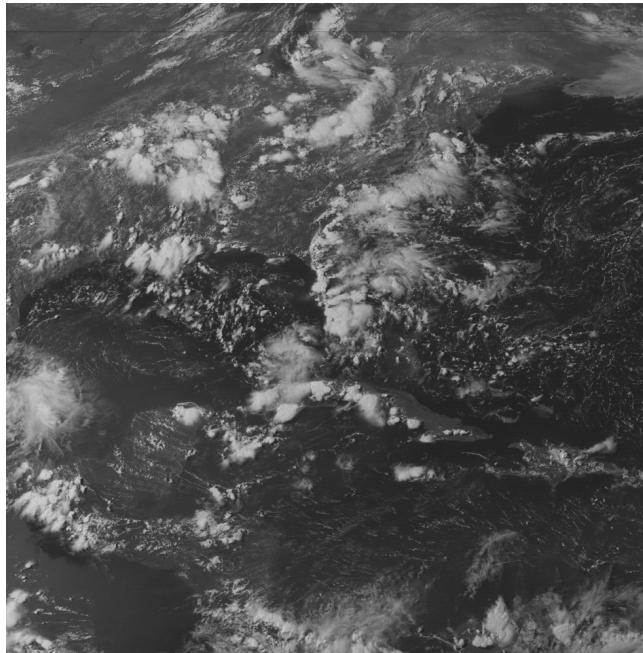
1. SMS1-VISSL\_L1-AOIPS-IR\_1974m0706t150519\_DD51982-2.TAP
2. SMS1-VISSL\_L1-AOIPS-IR\_1974m0706t151040\_DD52930-2.TAP
3. SMS1-VISSL\_L1-AOIPS-IR\_1974m0707t113640\_DD51983-2.TAP
4. SMS1-VISSL\_L1-AOIPS-IR\_1974m0710t103532\_DD52932-2.TAP
5. SMS1-VISSL\_L1-AOIPS-IR\_1975m0109t093152\_DD69593-1.TAP
6. SMS1-VISSL\_L1-AOIPS-IR\_1975m0111t000630\_DD69599-1.TAP
7. SMS1-VISSL\_L1-AOIPS-IR\_1979m0410t201428\_DD58022-2.TAP
8. SMS1-VISSL\_L1-AOIPS-IRG\_1974m0706t150519\_DD51982-3.TAP
9. SMS1-VISSL\_L1-AOIPS-IRG\_1974m0706t151040\_DD52930-3.TAP
10. SMS1-VISSL\_L1-AOIPS-IRG\_1974m0707t113640\_DD51983-3.TAP
11. SMS1-VISSL\_L1-AOIPS-IRG\_1974m0710t103532\_DD52932-3.TAP
12. SMS1-VISSL\_L1-AOIPS-IRG\_1975m0109t093152\_DD69593-2.TAP
13. SMS1-VISSL\_L1-AOIPS-IRG\_1975m0111t000630\_DD69599-2.TAP
14. SMS1-VISSL\_L1-AOIPS-IRG\_1979m0410t201428\_DD58022-3.TAP
15. SMS1-VISSL\_L1-AOIPS-VIS\_1974m0706t150519\_DD51982-1.TAP
16. SMS1-VISSL\_L1-AOIPS-VIS\_1974m0706t151040\_DD52930-1.TAP
17. SMS1-VISSL\_L1-AOIPS-VIS\_1974m0707t113640\_DD51983-1.TAP
18. SMS1-VISSL\_L1-AOIPS-VIS\_1974m0710t103532\_DD52932-1.TAP
19. SMS1-VISSL\_L1-AOIPS-VIS\_1979m0410t201428\_DD58022-1.TAP

The visible mode A images are 3904 pixels wide, IR and IRG images are 976 pixels wide, and the two IR/IRG global images are 3822 pixels wide. All of the recovered SMS 1 VISSR AOIPS files are archived at the GES DISC.

## 2.3 Key Science Data Fields

The primary science data fields in this data product are the VISSR IR and visible images stored as 8-bit bytes.

**Figure 1:** Typical SMS 1 VISSR Level 1 AOIPS data files showing a visible scene at left, and an infrared scene with grid (cool = white, warm = black) at right, of the Gulf of Mexico, Caribbean Sea and Western Atlantic region from September 1, 1975 at 21:02:43 to 21:07:43.



# 3. Data Contents

---

The granularity of this data product is typically around 20 minutes.

## 3.1 Data Records

The file format is described in various sections of Appendix B of the “VISSR Data Processing Plan for SMS/GOES Satellites” document.

The original tape files each included a tape header file. These were then followed by a set of up to 3 data files. As part of the recovery, the GES DISC has extracted and archived the data files from the tapes. There are 3 types of data files containing either an IR image, and IR grid, or a Visible image. Each data file includes two header label records which is followed by a set of data image records. The original data were written on IBM machines and use big-endian byte order. There were two types of tapes received. The first from University of Wisconsin SSEC contains record blocks preceded by an 8 BCD character indicating the record block size. The second from Redstone contain record blocks preceded by a 4-byte integer word with the record block size.

**Table 3-1-1: AOIPS Label Record (512 EBCDIC characters)**

Byte	Description	Size	Comments
1 – 8	Image Name	8 EBCDIC	first 4 characters is satellite (ME01)
9 – 16	Parent Name	8 EBCDIC	same as Image Name
17 – 24	Master Name	8 EBCDIC	same as Image Name
25 – 32	Tape Number	8 EBCDIC	6 digit picture number plus 2 digit DECOM reel number
33 – 36	User ID	4 EBCDIC	always IPDM
37 – 38	Start Date/Time of Scan Line 1 (Full Image)	16-bit int	(YY x 100) + MM
39 – 40		16-bit int	(DD x 100) + HH
41 – 42		16-bit int	(MM x 100) + SS
43 – 44		16-bit int	Milliseconds x 10
45 – 46	Access Code	16-bit int	always 0
47 – 48	Application ID	16-bit int	always 1
49 – 50	Data Set Reference Number	16-bit int	always 0
51 – 52	Sequence Number in Data Set	16-bit int	always 0
53 – 54	Label Type	16-bit int	always 0

55 - 56	File Format (Interleaving Code)	16-bit int	always 0
57 - 60	Exchange Format Identification Code	4 EBCDIC	always EFIC
61 - 62	Pixel Format	16-bit int	always 1
63 - 64	Image File Number	16-bit int	
65 - 66	Record Length	16-bit int	in 8-bit bytes
67 - 68	File Length	16-bit int	Max records, excluding this one
69 - 70	Left Edge Fill	16-bit int	8-bit bytes preceding first image pixel
71 - 72	Top Edge Fill	16-bit int	number of IPD and telemetry records
73 - 74	Number of Image Pixels	16-bit int	
75 - 76	Number of Image Lines	16-bit int	
77 - 78	Image Start Pixel	16-bit int	
79 - 80	Image Start Line	16-bit int	
81 - 82	Parent Start Pixel	16-bit int	same as Image Start Pixel
83 - 84	Parent Start Line	16-bit int	same as Image Start Line
85 - 86	Pixel Zoom Factor	16-bit int	always 1
87 - 88	Line Zoom Factor	16-bit int	always 1
89 - 90	Parent Pixel Scale (numerator)	16-bit int	always 1
91 - 92	Parent Pixel Scale (denominator)	16-bit int	always 1
93 - 94	Parent Line Scale (numerator)	16-bit int	always 1
95 - 96	Parent Line Scale (denominator)	16-bit int	always 1
97 - 98	Pixel Offset	16-bit int	always 0
99 - 100	Line Offset	16-bit int	always 0
101 - 106	Not Used	6 bytes	zero filled
107 - 108	Max Number of Secondary Data Records	16-bit int	always 9
109 - 110	Logical Record Length of Secondary Records	16-bit int	always 720
111 - 112	Line Documentation Length	16-bit int	follows image data in each record
113 - 118	Not Used	6 bytes	zero filled
119 - 120	Program ID	16-bit int	Always 1 for files by SMS VISSR DPP
121 - 122	Program Version Creation Date	16-bit int	(MM x 1000) + DD x 10 + Y
123 - 192	Not Used	70 bytes	zero filled
193 - 194	Band Code	16-bit int	1=VIS; 2=IR; 3=IR Grid; 4=Calibration
195 - 196	Pixel Width (1/2 mile units)	16-bit int	4=IR; 2=VIS Mode B; 1=VIS Mode A
197 - 198	Line Height (1/2 mile units)	16-bit int	8=IR; 4=VIS Mode B; 1=VIS Mode A
199 - 200	Number of Orbit/Attitude Blocks	16-bit int	always 0, 1 or 2
201 - 202	Number of File Containing IR Calibration Data	16-bit int	zero if IR calibration file not present
203 - 204	Scaling Table ID	16-bit int	

205 - 208	Degree Scale Factor (DSCL)	32-bit int	always $2^{16} = 65536$	
209 - 212	Kilometer Scale Factor (KSCL)	32-bit int	always $2^{11} = 2048$	
213 - 216	Date of Orbit/Attitude Data	32-bit int	(YY x 1000) + day	
217 - 220	Time of Orbit/Attitude Data	32-bit int	milliseconds	
221 - 224	Attitude ID	32-bit int		
225 - 228	Geodetic Latitude (deg; positive north)	32-bit int	scaled by DSCL	
229 - 232	Longitude (deg; positive east)	32-bit int	scaled by DSCL	
233 - 236	Height Above Oblate Earth (km)	32-bit int	scaled by KSCL	
237 - 240	Right Ascension (R.A.) of Spin Axis (deg)	32-bit int	scaled by DSCL	
241 - 244	Declination of Spin Axis (deg)	32-bit int	scaled by DSCL	
245 - 248	Spin Period (microseconds)	32-bit int		
249 - 252	R.A. of Position Vector w.r.t. Aries (deg)	32-bit int	scaled by DSCL	
253 - 256	X Cartesian Coordinate Position Vector (km)	32-bit int	scaled by KSCL	
257 - 260	Y "	32-bit int		
261 - 264	Z "	32-bit int		
265 - 268	X Cartesian Coordinate Velocity Vector (km/h)	32-bit int	scaled by KSCL	
269 - 272	Y "	32-bit int		
273 - 276	Z "	32-bit int		
277 - 340	Data for Second Block of Orbit/Attitude Data	16 x 32-bit int	Same Format as First Block Above	
341 - 512	Not Used	172 bytes	zero filled	

First Orbit/Attitude Block

**Table 3-1-2:** IPD Label Record (IR/IRG=1244/4092 byte or VIS=4172 byte VIS EBCDIC characters)

Byte	Description	Size	Comments
1 - 7	International Code (7403301 for SMS 1)	7	
9 - 14	Date of Recording (YYMMDD)	6	
16 - 18	Station Code	3	
20	Analog Tape Deck Identification	1	
22 - 27	Tape Number	6	
29 - 32	Start Time of Analog Tape (HHMM)	4	
34 - 37	Stop Time of Analog Tape (HHMM)	4	
39 - 44	Date of Digitization (YYMMDD)	6	
46	Picture Section Number	1	
48	Buffer Tape Reel Number	1	
50	Buffer Tape File Number	1	
52	Buffer Tape Section File Number	1	
54 - 55	Pass Two Line Identification	2	

57	Pre-Edit Tape Reel Number	1	
59	Pre-Edit Tape File Number	1	
61	Pre-Edit Tape Selection File Number	1	Section file is relative to first file
63	Master Data Tape Reel Number	1	
65	Master Data Tape File Number	1	
66 - 67	Master Data Tape Section File Number	2	Section file is relative to first file
69 - 71	Start Day of Year (3 digit)	3	
72 - 77	Start Time (HHMMSS)	6	
79 - 84	Stop Time (HHMMSS)	6	
86 - 89	Elapsed Time (MMSS)	4	
91	Data Mode	1	A = Visible Mode A B = Visible Mode B G = IR Grid I = IR Image K = IR Calibration
93	IR Data Only (Y or N)	1	
95 - 98	Initial Image Line Number	4	
100 - 103	Final Image Line Number	4	
105 - 109	DECOM Run Number	5	
111 - 112	DECOM Reel Number	2	
114	DECOM Reel File Number	1	
115 - 116	DECOM Section File Number	2	Section file is relative to first file
122	O/A/TM Data Present (Y or N)	1	
124	GMT Present From Time Track (Y, N or D)	1	Y = Valid GMT N = Flywheel GMT, D = from Documentation Time
126 - 130	Creation Date (YMMDD)	5	
132	Tape Identification	1	A = AOIPS Picture Tape
134 - 137	Start Pixel (Relative to IR)	4	
139	Sector Size Code	1	
141 - 144	Scaling Table ID	4	
145 - 149	Scaling Table Date (YYDDD)	5	
150 - END	Not Used		blank character padded

**Table 3-1-3:** Telemetry Record (1244/4092 bytes IR/IRG or 4172 bytes VIS)

Byte	Contents
1 - 720	16 telemetry frames of 40 9-bit words per frame packed in 45 8-bit bytes for a total of 640 9-bit words A full normal mode telemetry block contains 64 frames requiring 4 records. Up to 2 blocks per file.
720 - end	Zero filled

**Table 3-1-4:** Image Data Record (1244/4092 bytes IR/IRG or 4172 bytes VIS)

Byte	Size	Contents
1 - N	N Bytes	Image Data where N is Number of Pixels (see AOIPS label) (IR/IRG = 976; IR/IRG full = 3822; VIS Mode A = 3904)
N+1 - N+256	129 x 16-bit int	IR Documentation (See Table 3-1-5 below)
N+257 - N+258	16-bit int	Day of Year
N+259 - N+262	32-bit int	Milliseconds of Day
N+263 - N+268	6 bytes (48 bits)	Data Flags: F1 PreEdit Data Flags (12 bits) F2 MDT Data Flags (12 bits) F3 Image Line Number (12 bits) F4 Scan Line Number (12 bits)
N+269 - N+270		Pad (Only When Necessary to Make Record Length an Even Multiple of 32 Bits)

**Table 3-1-5:** IR Documentation

Bytes	Contents
1	Retrace <sup>1</sup> – one indicates scanner retrace
2	Spacecraft Name
3	Unused
4	Frame Code <sup>1</sup> – one indicates picture transmission
5	Change Code <sup>1</sup> – one indicates start of picture if frame code is one or end of picture if frame code is zero
6	Step Code <sup>1</sup> – one indicates normal line transmission; zero indicates that this is not to be used to expose film and facsimile recorder line is not to be incremented (stepped).
7	Line Delay – this number (1 – 8) denotes delay to be introduced by user; expressed in bit in intervals.
8	IR Selection – 000000001 = IR1, 000000010 = IR2, 000000100 = AVG
9	Gray Scale Status <sup>1</sup> – one indicates gray scale information retransmission
10	Direct Transmission Mode <sup>1</sup> – one indicates 28 Mb/sec; zero indicates 14 Mb/sec
11	Image Line Number – BCD value split into 2 characters/word 2 most significant BCD characters
12	2 least significant BCD characters
13	Mode Code – A = 001100100, B = 01011001, C = 010110010, D = 010001111

14	Beta Count	MSB	0	8 MSB	LSB
15			0	8 MSB	
16			0	8 LSB	
17	Grid/No Grid <sup>1</sup> – Zero indicates no grid information				
18	Sync Error	MSB		LSB	
		0	8 MSB		
19		0	7 LSB	0	
20	Bit Error Count -	MSB		LSB	
		0	8 MSB		
21		0	5 LSB	000	
22	Setup Error <sup>1</sup> – one indicates setup error				
23 - 24	Computer Error Message MSB      00000001      LSB 00000010 :            : 10000000				
25	Scan Count –	Two most significant BCD characters			
26		Two least significant BCD characters			
27	Year – 2 MSD in BCD characters				
28	Year – 2 LSD in BCD characters				
29	Day of Year – 2 MSD in BCD characters				
30	Day of Year – 2 MSD in BCD characters				
31	Hour – 2 BCD characters)				
32	Minute – 2 BCD characters				
33	Second – 2 BCD characters)				
34	Millisecond × 10 – 2 BCD characters				
35	Black Enable <sup>1</sup> – one indicates annotation transmission				
36	Mode C-Calibrate <sup>1</sup> – one indicates C-Cal is not used; otherwise: MSB V <sub>1</sub> 00000001 V <sub>2</sub> 00000010 V <sub>3</sub> 00000100 V <sub>4</sub> 00001000 V <sub>5</sub> 00010000 V <sub>6</sub> 00100000 V <sub>7</sub> 01000000 V <sub>8</sub> 10000000				
37	Bit/Frame Sync Lock	MSB	LSB	00000001X	Bit Lock      1      loss
				00000010X	Frame Lock      X = 1 for any
				00000100X	Bit Freq Lock      of lock
38	Limited Scan Mode Indicator <sup>1</sup> – one indicates limited scan mode				

39	Sample Control Mode (LSB)	IR – 2 PT IR – 1 PT IR – EMP VIS – 4 PT VIS – 2 PT VIS – 1 PT VIS – EMT
40	Visible Channel Connection (LSB)	V1 V2 V3 Coding for each channel is V4 as follows: V5 0 = normal V6 1 = patched input is used V7 V8
41	Scan Direction <sup>1</sup>	– one indicates normal North-South direction (may not be used)
42	Bi-phase Modulator On/Off <sup>1</sup>	– one indicates modulator is on
43	Unused	
44	PLL Error Light <sup>1</sup>	– one indicates error condition
45	Test Data	MSB      LSB 000000000      Normal 000000001      Local 000000010      Remote 000000100      Comp Gen IR
46	Data Randomization <sup>1</sup>	– one indicates on
47	Sun Pulse Select <sup>1</sup>	– one indicates digital, zero indicates analog
48	Ness Mode Select	MSB      LSB 000000000      4 x 4 IR 000000001      Max SV 000000010      4 x 2 IR
49	Limited Scan Command Encoder Enable <sup>1</sup>	– one indicates on
50	Digital Sun Pulse	– 8 LSB's contain the digital sun pulse
51	Bit Error Light <sup>1</sup>	– one indicates on
52	Mean IR Difference	
53	RMS IR Difference	
54 – 55	Correction Table ID	
56	Left Horizon Point – 8 MSB	
57		8 LSB
58	Right Horizon Point – 8 MSB	
59		8 LSB
60	Equatorial Scan Count – 8 MSB	
61		8 LSB

62 - 79	Unused																																								
80	<p>Telemetry Code Word</p> <table style="margin-left: 100px;"> <tr> <td style="text-align: left;">MSB</td> <td style="text-align: right;">LSB</td> </tr> <tr> <td style="text-align: center;">0 A B C D E F G H</td> <td></td> </tr> </table> <p>A and B      Unused; always zero</p> <p>C = D = 0      Indicates frame sync search; no valid telemetry, three or more bad sync patterns since lock</p> <p>C = 0, D = 1    Indicates frame sync in check1; one good sync since search or two bad syncs since lock</p> <p>C = 1, D = 0    Indicates frame sync in check2; two good sync since search or one bad sync since lock</p> <p>C = D = 1      Indicates frame sync in lock; three or more good sync patterns found</p> <p>EFG = 0-3      Telemetry quarter-frame number</p> <ul style="list-style-type: none"> <li>4      First output without telemetry</li> <li>5      Second output without telemetry</li> <li>6      Third output without telemetry</li> <li>7      Fourth or greater output without telemetry</li> </ul> <p>H      Parity bit, exclusive or of B, D and F</p>	MSB	LSB	0 A B C D E F G H																																					
MSB	LSB																																								
0 A B C D E F G H																																									
81 - 98	One quarter of telemetry frame; 16 9-bit words packed into the 8 LSBs of each 9-bit IR doc word																																								
99	<p>O/A block number (1 - 10)</p> <p>One quarter of an orbit/attitude data block; contents are as follows<sup>2</sup>:</p> <table style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 25%;">First <u>Quarter</u></th> <th style="width: 25%;">Second <u>Quarter</u></th> <th style="width: 25%;">Third <u>Quarter</u></th> <th style="width: 25%;">Fourth <u>Quarter</u></th> </tr> </thead> <tbody> <tr> <td>100</td> <td>1</td> <td>2</td> <td>3</td> </tr> <tr> <td>101 - 104</td> <td>Date (YYDD)</td> <td>X Position Vector (km) x 2<sup>13</sup></td> <td>Spin Period (μsec)</td> </tr> <tr> <td>105 - 108</td> <td>GMT Time (ms) x 10</td> <td>Y Position Vector (km) x 2<sup>13</sup></td> <td>Spin Axis RA (deg) x 2<sup>21</sup></td> </tr> <tr> <td>109 - 112</td> <td>RA from 0° Meridian</td> <td>Z Position Vector (km) x 2<sup>13</sup></td> <td>Spin Axis Dec (deg) x 2<sup>21</sup></td> </tr> <tr> <td>113 - 116</td> <td>S/C Lat (deg) x 2<sup>21</sup></td> <td>X' Velocity Vector(km/h) x 2<sup>13</sup></td> <td>Pitch (deg) x 2<sup>21</sup> (odd block) N-S Step Angle 2<sup>21</sup> (even block)</td> </tr> <tr> <td>117 - 120</td> <td>S/C Lon (deg) x 2<sup>21</sup></td> <td>Y' Velocity Vector(km/h) x 2<sup>13</sup></td> <td>Roll (deg) x 2<sup>21</sup> (odd block) E-W Step Angle 2<sup>21</sup> (even block)</td> </tr> <tr> <td>121 - 124</td> <td>S/C Height (km) x 2<sup>13</sup></td> <td>Z' Velocity Vector(km/h) x 2<sup>13</sup></td> <td>           Yaw (deg) x 2<sup>21</sup> (odd block)            Attitude ID:            0 = other method            1 = sun/earth sensor            2 = landmark (2-mile imagery)            3 = PICATT (earth edge data)            4 = etc., for later use         </td> </tr> <tr> <td>125 - 127</td> <td>Unused</td> <td></td> <td></td> </tr> <tr> <td>128</td> <td>Longitudinal Parity Check (ODD)</td> <td></td> <td></td> </tr> </tbody> </table>	First <u>Quarter</u>	Second <u>Quarter</u>	Third <u>Quarter</u>	Fourth <u>Quarter</u>	100	1	2	3	101 - 104	Date (YYDD)	X Position Vector (km) x 2 <sup>13</sup>	Spin Period (μsec)	105 - 108	GMT Time (ms) x 10	Y Position Vector (km) x 2 <sup>13</sup>	Spin Axis RA (deg) x 2 <sup>21</sup>	109 - 112	RA from 0° Meridian	Z Position Vector (km) x 2 <sup>13</sup>	Spin Axis Dec (deg) x 2 <sup>21</sup>	113 - 116	S/C Lat (deg) x 2 <sup>21</sup>	X' Velocity Vector(km/h) x 2 <sup>13</sup>	Pitch (deg) x 2 <sup>21</sup> (odd block) N-S Step Angle 2 <sup>21</sup> (even block)	117 - 120	S/C Lon (deg) x 2 <sup>21</sup>	Y' Velocity Vector(km/h) x 2 <sup>13</sup>	Roll (deg) x 2 <sup>21</sup> (odd block) E-W Step Angle 2 <sup>21</sup> (even block)	121 - 124	S/C Height (km) x 2 <sup>13</sup>	Z' Velocity Vector(km/h) x 2 <sup>13</sup>	Yaw (deg) x 2 <sup>21</sup> (odd block) Attitude ID: 0 = other method 1 = sun/earth sensor 2 = landmark (2-mile imagery) 3 = PICATT (earth edge data) 4 = etc., for later use	125 - 127	Unused			128	Longitudinal Parity Check (ODD)		
First <u>Quarter</u>	Second <u>Quarter</u>	Third <u>Quarter</u>	Fourth <u>Quarter</u>																																						
100	1	2	3																																						
101 - 104	Date (YYDD)	X Position Vector (km) x 2 <sup>13</sup>	Spin Period (μsec)																																						
105 - 108	GMT Time (ms) x 10	Y Position Vector (km) x 2 <sup>13</sup>	Spin Axis RA (deg) x 2 <sup>21</sup>																																						
109 - 112	RA from 0° Meridian	Z Position Vector (km) x 2 <sup>13</sup>	Spin Axis Dec (deg) x 2 <sup>21</sup>																																						
113 - 116	S/C Lat (deg) x 2 <sup>21</sup>	X' Velocity Vector(km/h) x 2 <sup>13</sup>	Pitch (deg) x 2 <sup>21</sup> (odd block) N-S Step Angle 2 <sup>21</sup> (even block)																																						
117 - 120	S/C Lon (deg) x 2 <sup>21</sup>	Y' Velocity Vector(km/h) x 2 <sup>13</sup>	Roll (deg) x 2 <sup>21</sup> (odd block) E-W Step Angle 2 <sup>21</sup> (even block)																																						
121 - 124	S/C Height (km) x 2 <sup>13</sup>	Z' Velocity Vector(km/h) x 2 <sup>13</sup>	Yaw (deg) x 2 <sup>21</sup> (odd block) Attitude ID: 0 = other method 1 = sun/earth sensor 2 = landmark (2-mile imagery) 3 = PICATT (earth edge data) 4 = etc., for later use																																						
125 - 127	Unused																																								
128	Longitudinal Parity Check (ODD)																																								

Note 1 All but the last bit in each code is identical, e.g., 00000001 (zero) or 11111110 (one)

Note 2 Regarding O/A data blocks, however words 100 to 124:

- a) If no O/A is present, word 100 will be zero
- b)  $B_\phi$ ,  $\dot{B}$ ,  $\tau_\phi$  are the coefficients of the equation for beta count, words 14-16.

The units of  $\dot{B}$  are counts per milliseconds,

and the second bit of  $\dot{B}$  is set to 1 to indicate a negative value.

## 3.2 Metadata

The metadata are contained in a separate XML formatted file having the same name as the data file with .xml appended to

**Table 3-2:** Metadata attributes associated with the data file.dad

Name	Description
LongName	Long name of the data product.
ShortName	Short name of the data product.
VersionID	Product or collection version.
GranuleID	Granule identifier, i.e. the name of the file.
Format	File format of the data file.
CheckSumType	Type of checksum used.
CheckSumValue	The value of the calculated checksum.
SizeBytesDataGranule	Size of the file or granule in bytes.
InsertDateTime	Date and time when the granule was inserted into the archive. The format for date is YYYY-MM-DD and time is hh-mm-ss.
RangeBeginningDate	Begin date when the data was collected in YYYY-MM-DD format.
RangeBeginningTime	Begin time of the date when the data was collected in hh-mm-ss format.
RangeEndingDate	End date when the data was collected in YYYY-MM-DD format.
RangeEndingTime	End time of the date when the data was collected in hh-mm-ss format.
PlatformShortName	Short name or acronym of the platform or satellite
InstrumentShortName	Short name or acronym of the instrument
SensorShortName	Short name or acronym of the sensor
Gpolygon:	Latitudes of the polygon points that represent the satellite coverage. Each point is identified by its latitude and longitude pair.
PointLatitude	
Gpolygon:	Longitudes of the polygon points that represent the satellite coverage. Each point is identified by its latitude and longitude pair.
PointLongitude	
Recovering_Institute_Tape_ID	The Tape ID used by the data recovery institute
ElapsedMinTime	Duration in minutes of data collected during an orbit.

## 4. Reading the Data

---

The data are written in a binary record-oriented format. Using the record format specification in the section above, users can write software to read the data files. Please note that the data were originally written using a big-endian order, and characters use the old IBM EBCDIC encoding, thus users will have to reconstruct the data values accordingly.

A sample FORTRAN program is included in the Appendix section which will read in the data records. Additionally a FORTRAN function is included to perform byte swapping.

## 5. Data Services

---

### 5.1 GES DISC Search

The GES DISC provides a keyword, spatial, temporal and advanced (event) searches through its unified search and download interface:

<https://disc.gsfc.nasa.gov/>

### 5.2 Documentation

The data product landing pages provide information about these data products, as well as links to download the data files and relevant documentation:

[https://disc.gsfc.nasa.gov/datacollection/VISSRSMS1L1AOIPS\\_001.html](https://disc.gsfc.nasa.gov/datacollection/VISSRSMS1L1AOIPS_001.html)

### 5.3 Direct Download

These data products are available for users to download directly using HTTPS:

<https://acdisc.gesdisc.eosdis.nasa.gov/data/SMS/VISSRSMS1L1AOIPS.001/>

## 6. More Information

---

### 6.1 Contact Information

Name: GES DISC Help Desk

URL: <https://disc.gsfc.nasa.gov/>

E-mail: [gsfc-help-disc@lists.nasa.gov](mailto:gsfc-help-disc@lists.nasa.gov)

Phone: 301-614-5224

Fax: 301-614-5228

Address: Goddard Earth Sciences Data and Information Services Center

Attn: Help Desk

Code 610.2

NASA Goddard Space Flight Center

Greenbelt, MD 20771, USA

### 6.2 References

"The GOES/SMS User's Guide", ed. Corbell R., Callahan, C, Kotsch, W., NOAA National Environmental Satellite Services, 1976

"VISSR Data Processing Plan for SMS/GOES Satellites" by McCowan, P. L., NASA Goddard Space Flight Center, September 1977

## **7. Appendices**

---

### **Acknowledgments**

The data recovery task at the GES DISC is funded by NASA's Earth Science Data and Information System program.

### **Acronyms**

*AOIPS*: Atmospheric and Oceanographic Information Processing System

*EOS*: Earth Observing System

*ESDIS*: Earth Science and Data Information System

*GES DISC*: Goddard Earth Sciences Data and Information Services Center

*GSFC*: Goddard Space Flight Center

*L1*: Level-1 Data

*NASA*: National Aeronautics and Space Administration

*QA*: Quality Assessment

*SMS*: Synchronous Meteorological Satellite

*VISSR*: Visible Infrared Spin-Scan Radiometer

# FORTRAN Code

```
C-----  
C ^NAME: READ_SMSAOIPS  
C   This program will read an SMS VISSR AOIPS Level-1 data file.  
C  
C   The SMS VISSR AOIPS files contain a header record of size 512 bytes,  
C   followed by a series of data records of size 1244 bytes (IR scenes),  
C   4092 bytes (IR global), or 4172 bytes (Visible Mode A). The data files  
C   will contain either VIS scan lines, IR scans, or the grids for IR scans  
C   for either a section of the earth, or full earth. Scan records contain  
C   the IR or visible image, as well as time, and other ancillary values.  
C   This program will print the contents of each record.  
  
C Please note the record size markers can vary. For SSEC recovered  
C files it is an 8 BCD character value at the beginning of the record.  
C For Redstone recovered files it is a 4-byte integer at the beginning and  
C end of the record. This program checks for both.  
C  
C ^MAJOR VARIABLES:  
C   FNAME - name of input file  
C   BUFF - buffer for data record  
C   TEMP - buffer for holding temporary 4-byte word  
C   WORD - integer 4-byte word  
C   IBLKSZ - size of record block in bytes  
C   IOS - I/O status number  
C  
C ^NOTES:  
C   Compile: gfortran -o READ_SMSAOIPS.EXE READ_SMSAOIPS.FOR  
C  
C ^ORGANIZATION: NASA/GSFC, Code 610.2  
C  
C ^AUTHOR: James Johnson  
C  
C ^ADDRESS: james.johnson@nasa.gov  
C  
C ^CREATED: Apr 02, 2020  
C-----
```

CHARACTER	FNAME*1024	! Filename
CHARACTER	BUFF(6000)	! Buffer for data record (max 4172)
CHARACTER	CHKIPD*7	! IPD Label check
INTEGER*4	IBLKSZ	! Size of records
INTEGER*4	IWORD	! 4-byte word
LOGICAL*1	RDFLAG	! Redstone Flag
CHARACTER	SSEC*8	! Buffer to hold 8-byte SSEC word
CHARACTER	TEMP(4)	! Buffer to hold 4-byte word
EQUIVALENCE	(TEMP,IWORD)	

```
C   Get the name of the input data file to read  
WRITE (0, *), 'Enter the name of the input file:'  
READ (5,'(A)') FNAME  
PRINT '("File = ",A60)', FNAME
```

```

C      Open the specified input file
OPEN (UNIT=1, FILE=FNAME, STATUS='OLD', ACCESS='DIRECT',
&      FORM='UNFORMATTED', RECL=1, ERR=99, IOSTAT=IOS)

C      Initialize N (record number), ND (data record), NT (telem record)
C      and IOFF (byte offset in file)
N=0
ND=0
NT=0
IOFF=0

C      Loop through the file reading all records in file
10 DO

C      Read the first 4-byte word or record size header
DO I=1,4
    READ (1, REC=IOFF+I, IOSTAT=IOS, ERR=90) TEMP(I)
    SSEC(I:I) = TEMP(I)
END DO
IOFF=IOFF+(I-1)

C      Check if this is a Univ Wisc SSEC tape file or one from Redstone
IF ((ICHAR(TEMP(1)).GE.48.AND.ICHAR(TEMP(1)).LE.57).AND.
+     (ICHAR(TEMP(2)).GE.48.AND.ICHAR(TEMP(2)).LE.57).AND.
+     (ICHAR(TEMP(3)).GE.48.AND.ICHAR(TEMP(3)).LE.57).AND.
+     (ICHAR(TEMP(4)).GE.48.AND.ICHAR(TEMP(4)).LE.57)) THEN
    DO I=1,4
        READ (1, REC=IOFF+I, IOSTAT=IOS, ERR=90) TEMP(I)
        SSEC(I+4:I+4) = TEMP(I)
    END DO
    IOFF=IOFF+(I-1)
    READ (SSEC,'(I8)') IBLKSZ                               ! SSEC File
    RDFLAG = .FALSE.
ELSE
    IBLKSZ = IWORD                                         ! Redstone File
    RDFLAG = .TRUE.
END IF

C      End-of-File (EOF) mark, goto quit
IF (IBLKSZ .EQ. 0) GOTO 90

C      Next read the block of data
DO I=1,IBLKSZ
    READ (1, REC=IOFF+I, IOSTAT=IOS) BUFF(I)
    IF (IOS .NE. 0) THEN
        PRINT '("ERROR: BUFF ",I4,X,I4,", IOSTAT: ",I6)', N,I-1,IOS
        IBLKSZ = I-1
        GOTO 90
    END IF
END DO
IOFF=IOFF+(I-1)
N = N+1

```

```

C      Determine Record Type
IF (N.EQ.1.OR.IBLKSZ.LE.512) THEN                                ! AOIPS Label
  DO I=IBLKSZ,512
    BUFF(I) = CHAR(64)
  END DO
  CALL PRAOIP(BUFF(1:512), IRECLN, NPIX)
ELSE
  IF (IRECLN.NE.1244.AND.IRECLN.NE.4092.AND.IRECLN.NE.4172) THEN
    IF (IBLKSZ.LE.1244) THEN                                         ! Fix IR reclen
      IRECLN = 1244
      NPIX = 976
    ELSE IF (IBLKSZ.GT.1244) THEN                                     ! Fix VIS reclen
      IRECLN = 4172
      NPIX = 3904
    END IF
  ENDIF

C      First 7 Chars of IPD Label should be IntlCode, Last Chars Blanks
DO I = 1,7
  CHKIPD(I:I) = CHAR(IEBC(ICHAR(BUFF(I))))                         ! EBCDIC text
END DO
ML = 0
DO I=150,IBLKSZ
  IF (ICHAR(BUFF(I)).EQ.64) THEN                                       ! Blank char
    ML = ML+1
  END IF
END DO

C      Check if IPD Label
IF (N.EQ.2.AND.(CHKIPD.EQ.'7403301'                                ! IPD Label
+ .OR.ML.EQ.(IBLKSZ-149))) THEN
  DO I=IBLKSZ,149
    BUFF(I) = CHAR(64)
  END DO
  CALL PRIPDL(BUFF(1:149))
ELSE
  DO I=IBLKSZ,IRECLN                                              ! Pad short recs
    BUFF(I) = CHAR(0)
  END DO
  MT = 0
  DO I=721,IRECLN
    IF (ICHAR(BUFF(I)).EQ.0) THEN                                     ! Null char
      MT = MT+1
    END IF
  END DO
END DO

C      Check if Telemetry or Image Record
IF (MT.EQ.IRECLN-720) THEN                                           ! Telem Record
  NT = NT+1
  CALL PRTREC(BUFF(1:720), NT)
ELSE
  ND = ND+1
  CALL PRIREC(BUFF(1:IRECLN), IRECLN, NPIX, ND)
END IF
END IF

```

```

        IF (RDFLAG.EQV..TRUE.) THEN                                ! Redstone uses
          DO I=1,4                                              ! end RecSize
            READ (1, REC=IOFF+I, IOSTAT=IOS, ERR=90) TEMP(I)
          END DO
          IOFF=IOFF+(I-1)
        END IF

      END DO

C Close the input file
  90 CLOSE(1)
  GOTO 100

  99 PRINT '("ERROR: OPEN FILE, IOSTAT: ",I6)', IOS

100 STOP
END

C-----
C      This Subroutine will print the AOIPS Label Record
C-----
SUBROUTINE PRAOIP(BUFF, IRECLN, NPIX)

CHARACTER      BUFF(512)                                ! Buffer for record block
CHARACTER      ASC*512                                 ! Buffer for ascii label
INTEGER*2       I2SWAP                                 ! Function swaps short ints
INTEGER*4       I4SWAP                                 ! Function swaps long ints

DO I = 1,512
  ASC(I:I) = CHAR(IEBC(ICHAR(BUFF(I))))           ! EBCDIC encoded text
END DO

PRINT '("===== AOIPS LABEL =====")'
PRINT '("ImageName  =",X,A8)', ASC(1:8)
PRINT '("ParentName =",X,A8)', ASC(9:16)
PRINT '("MasterName =",X,A8)', ASC(17:24)
PRINT '("TapeNumber =",X,A8)', ASC(25:32)
PRINT '("UserID     =",X,A4)', ASC(33:36)
PRINT '("Scan1Date  =",4(X,I4.4))', I2SWAP(BUFF( 37:38)),
+ I2SWAP(BUFF( 39:40)), I2SWAP(BUFF( 41:42)), I2SWAP(BUFF( 43:44))
PRINT '("AccessCode  =",I5)', I2SWAP(BUFF( 45:46))
PRINT '("ApplicatID  =",I5)', I2SWAP(BUFF( 47:48))
PRINT '("DsetRefNum =",I5)', I2SWAP(BUFF( 49:50))
PRINT '("SeqNum     =",I5)', I2SWAP(BUFF( 51:52))
PRINT '("LabelType   =",I5)', I2SWAP(BUFF( 53:54))
PRINT '("FileFormat  =",I5)', I2SWAP(BUFF( 55:56))
PRINT '("ExchFormat  =",A4)', ASC(57:60)
PRINT '("PixelFormat =",I5)', I2SWAP(BUFF( 61:62))
PRINT '("ImgFileNum  =",I5)', I2SWAP(BUFF( 63:64))
IRECLN = I2SWAP(BUFF( 65:66))
PRINT '("RecLength   =",I5)', IRECLN
PRINT '("NumRecords  =",I5)', I2SWAP(BUFF( 67:68))
PRINT '("LftEdgFill  =",I5)', I2SWAP(BUFF( 69:70))
PRINT '("TopEdgFill  =",I5)', I2SWAP(BUFF( 71:72))
NPIX  = I2SWAP(BUFF( 73:74))
PRINT '("NumPixels   =",I5)', NPIX
PRINT '("NumLines    =",I5)', I2SWAP(BUFF( 75:76))

```

```

PRINT '("IStartPixel = ",I5)', I2SWAP(BUFF( 77:78))
PRINT '("IStartLine = ",I5)', I2SWAP(BUFF( 79:80))
PRINT '("PStartPixel = ",I5)', I2SWAP(BUFF( 81:82))
PRINT '("PStartLine = ",I5)', I2SWAP(BUFF( 83:84))
PRINT '("PixelZoom = ",I5)', I2SWAP(BUFF( 85:86))
PRINT '("LineZoom = ",I5)', I2SWAP(BUFF( 87:88))
PRINT '("PixelSclNmr = ",I5)', I2SWAP(BUFF( 89:90))
PRINT '("PixelSclDen = ",I5)', I2SWAP(BUFF( 91:92))
PRINT '("LineSclNmr = ",I5)', I2SWAP(BUFF( 93:94))
PRINT '("LineSclDen = ",I5)', I2SWAP(BUFF( 95:96))
PRINT '("PixelOffset = ",I5)', I2SWAP(BUFF( 97:98))
PRINT '("LineOffset = ",I5)', I2SWAP(BUFF( 99:100))
C PRINT '("Unused1 = ",3(X,I5))', I2SWAP(BUFF(101:102)),
C + I2SWAP(BUFF(103:104)), I2SWAP(BUFF(105:106)),
  PRINT '("Max2ryRecs = ",I5)', I2SWAP(BUFF(107:108))
  PRINT '("Len2ryRecs = ",I5)', I2SWAP(BUFF(109:110))
  PRINT '("LineDocLen = ",I5)', I2SWAP(BUFF(111:112))
C PRINT '("Unused2 = ",3(X,I5))', I2SWAP(BUFF(113:114)),
C + I2SWAP(BUFF(115:116)), I2SWAP(BUFF(117:118)),
  PRINT '("ProgID = ",I5)', I2SWAP(BUFF(119:120))
  PRINT '("ProgVerID = ",I5)', I2SWAP(BUFF(121:122))
C PRINT '("Unused3 = ",35(X,I5))', I2SWAP(BUFF(123:124)),
C + I2SWAP(BUFF(125:126)), I2SWAP(BUFF(127:128)),
C +
C + I2SWAP(BUFF(189:190)), I2SWAP(BUFF(191:192))
  PRINT '("BandCode = ",I5)', I2SWAP(BUFF(193:194))
  PRINT '("PixelWidth = ",I5)', I2SWAP(BUFF(195:196))
  PRINT '("LineHeight = ",I5)', I2SWAP(BUFF(197:198))
  PRINT '("NumOABlks = ",I5)', I2SWAP(BUFF(199:200))
  PRINT '("FileIRCal = ",I5)', I2SWAP(BUFF(201:202))
  PRINT '("ScaleTable = ",I5)', I2SWAP(BUFF(203:204))
  PRINT '("DSCL = ",I5)', I4SWAP(BUFF(205:208))
  PRINT '("KSCL = ",I5)', I4SWAP(BUFF(209:212))
DO I=0,1
  PRINT '("----- OA Block ",I1," -----")', I+1
  PRINT '("Date = ",I10)', I4SWAP(BUFF(213+I*64:216+I*64))
  PRINT '("Time = ",I10)', I4SWAP(BUFF(217+I*64:220+I*64))
  PRINT '("AttID = ",I10)', I4SWAP(BUFF(221+I*64:224+I*64))
  PRINT '("Lat = ",I10)', I4SWAP(BUFF(225+I*64:228+I*64))
  PRINT '("Lon = ",I10)', I4SWAP(BUFF(229+I*64:232+I*64))
  PRINT '("Height = ",I10)', I4SWAP(BUFF(233+I*64:236+I*64))
  PRINT '("RASpinAxs = ",I10)', I4SWAP(BUFF(237+I*64:240+I*64))
  PRINT '("DecSpinAxs = ",I10)', I4SWAP(BUFF(241+I*64:244+I*64))
  PRINT '("SpinPeriod = ",I10)', I4SWAP(BUFF(245+I*64:248+I*64))
  PRINT '("RAPosVect = ",I10)', I4SWAP(BUFF(249+I*64:252+I*64))
  PRINT '("XPosVect = ",I10)', I4SWAP(BUFF(253+I*64:256+I*64))
  PRINT '("YPosVect = ",I10)', I4SWAP(BUFF(257+I*64:260+I*64))
  PRINT '("ZPosVect = ",I10)', I4SWAP(BUFF(261+I*64:264+I*64))
  PRINT '("XVelVect = ",I10)', I4SWAP(BUFF(265+I*64:268+I*64))
  PRINT '("YVelVect = ",I10)', I4SWAP(BUFF(269+I*64:272+I*64))
  PRINT '("ZVelVect = ",I10)', I4SWAP(BUFF(273+I*64:276+I*64))
END DO

RETURN
END

```

```

C-----  

C      This Subroutine will print the IPD Label Record  

C-----  

SUBROUTINE PRIPDL(BUFF)  

CHARACTER      BUFF(149)                      ! Buffer for record block  

CHARACTER      ASC*149                         ! Buffer for ascii label  

DO I = 1,149  

  ASC(I:I) = CHAR(IEBC(ICHAR(BUFF(I))))       ! EBCDIC encoded text  

END DO  

PRINT '("===== IPD LABEL =====")'  

PRINT '("IntlCode    = ",A7)', ASC(1:7)  

PRINT '("DateRecd   = ",A6)', ASC(9:14)  

PRINT '("Station     = ",A3)', ASC(16:18)  

PRINT '("TapeDeckID  = ",A1)', ASC(20:20)  

PRINT '("TapeNum     = ",A6)', ASC(22:27)  

PRINT '("TapeStart   = ",A4)', ASC(29:32)  

PRINT '("TapeStop    = ",A4)', ASC(34:37)  

PRINT '("DigitzDate  = ",A6)', ASC(39:44)  

PRINT '("PicSectNo   = ",A1)', ASC(46:46)  

PRINT '("BuffReelNo  = ",A1)', ASC(48:48)  

PRINT '("BuffFileNo  = ",A1)', ASC(50:50)  

PRINT '("BuffSectNo  = ",A1)', ASC(52:52)  

PRINT '("Pass2Line   = ",A2)', ASC(54:55)  

PRINT '("PredReelNo  = ",A1)', ASC(57:57)  

PRINT '("PredFileNo  = ",A1)', ASC(59:59)  

PRINT '("PredSectNo  = ",A1)', ASC(61:61)  

PRINT '("MDTReelNo   = ",A1)', ASC(63:63)  

PRINT '("MDTFileNo   = ",A1)', ASC(65:65)  

PRINT '("MDTSectNo   = ",A1)', ASC(66:67)  

PRINT '("StartDay    = ",A3)', ASC(69:71)  

PRINT '("StartTime   = ",A6)', ASC(72:77)  

PRINT '("StopTime    = ",A6)', ASC(79:84)  

PRINT '("ElapTime    = ",A4)', ASC(86:89)  

PRINT '("DataMode    = ",A1)', ASC(91:91)  

PRINT '("IRDataFlag  = ",A1)', ASC(93:93)  

PRINT '("InitLineNo  = ",A4)', ASC(95:98)  

PRINT '("LastLineNo  = ",A4)', ASC(100:103)  

PRINT '("DECOMRun    = ",A5)', ASC(105:109)  

PRINT '("DECOMReel   = ",A2)', ASC(111:112)  

PRINT '("DECOMFile   = ",A1)', ASC(114:114)  

PRINT '("DECOMSect   = ",A2)', ASC(115:116)  

PRINT '("PctDatRcov = ",A3)', ASC(118:120)  

PRINT '("TelemFlag   = ",A1)', ASC(122:122)  

PRINT '("GMTFlag     = ",A1)', ASC(124:124)  

PRINT '("CreateDate  = ",A5)', ASC(126:130)  

PRINT '("TapeID      = ",A1)', ASC(132:132)  

PRINT '("StartPixel  = ",A4)', ASC(134:137)  

PRINT '("SectSzCode  = ",A1)', ASC(139:139)  

PRINT '("SclTblID    = ",A4)', ASC(141:144)  

PRINT '("SclTblDate  = ",A5)', ASC(145:149)  

RETURN  

END

```

```

C-----  

C      This Subroutine will print the Telemetry Data Record  

C-----  

SUBROUTINE PRTREC(BUFF, M)  

CHARACTER      BUFF(720)                      ! Buffer for Record Block  

INTEGER*2       TFRAME(40,16)                   ! Telemetry Frame Array  

C      THERE ARE 16 TELEMETRY FRAMES OF 40 9-BIT WORDS PER FRAME  

DO N=1,720,9  

    I = (N-1)/45+1  

    J = MOD(N/9*8, 40)+1  

    TFRAME(J+0,I) = ISHFT(IAND(ICHAR(BUFF(N+0)), 'FF'Z),1)+  

+           ISHFT(ICHAR(BUFF(N+1)), -7)  

    TFRAME(J+1,I) = ISHFT(IAND(ICHAR(BUFF(N+1)), '7F'Z),2)+  

+           ISHFT(ICHAR(BUFF(N+2)), -6)  

    TFRAME(J+2,I) = ISHFT(IAND(ICHAR(BUFF(N+2)), '3F'Z),3)+  

+           ISHFT(ICHAR(BUFF(N+3)), -5)  

    TFRAME(J+3,I) = ISHFT(IAND(ICHAR(BUFF(N+3)), '1F'Z),4)+  

+           ISHFT(ICHAR(BUFF(N+4)), -4)  

    TFRAME(J+4,I) = ISHFT(IAND(ICHAR(BUFF(N+4)), '0F'Z),5)+  

+           ISHFT(ICHAR(BUFF(N+5)), -3)  

    TFRAME(J+5,I) = ISHFT(IAND(ICHAR(BUFF(N+5)), '07'Z),6)+  

+           ISHFT(ICHAR(BUFF(N+6)), -2)  

    TFRAME(J+6,I) = ISHFT(IAND(ICHAR(BUFF(N+6)), '03'Z),7)+  

+           ISHFT(ICHAR(BUFF(N+7)), -1)  

    TFRAME(J+7,I) = ISHFT(IAND(ICHAR(BUFF(N+7)), '01'Z),8)+  

+           ISHFT(ICHAR(BUFF(N+8)), -0)  

END DO  

PRINT '("===== TELEM REC ",I2," =====")', M  

DO I=1,16  

    PRINT '("Frame ",I2," =",17(X,I3))', I, TFRAME(1:17,I)  

    PRINT '(20(X,I3))', TFRAME(18:40,I)  

END DO  

RETURN  

END  

C-----  

C      This Subroutine will print the Image Data Record  

C-----  

SUBROUTINE PRIREC(BUFF, IRECLN, NPIX, M)  

CHARACTER      BUFF(IRECLN)                    ! Buffer for record block  

INTEGER*2       IMADATA(NPIX)                  ! Image Scan line data  

INTEGER*2       IDFLAG(4)                     ! Data Flags  

INTEGER*2       I2SWAP                       ! Function swaps short ints  

INTEGER*4       I4SWAP                       ! Function swaps long ints  

C      No. Pixels depends on IR/IRG(normal)=976, IR/IRG(global)=3822, or VIS=3904  

PRINT '("===== IMAGE REC ",I4," =====")', M  

PRINT '("Pixel Data = ")'  

DO N=1,NPIX  

    IMADATA(N) = ICHAR(BUFF(N))  

END DO  

PRINT '(16(X,I4))', IMADATA

```

```

C   IR DOCUMENTATION
IF (ISHFT(ICHAR(BUFF(NPIX+1)), -7).EQ.1) THEN
  CALL PRIDOC(BUFF(NPIX+1:NPIX+256))
END IF

C   GMT FLYWHEEL TIME
PRINT '("DayOfYear  = ",I8)', I2SWAP(BUFF(NPIX+256+1:NPIX+256+2))
PRINT '("MilliSecs  = ",I8)', I4SWAP(BUFF(NPIX+258+1:NPIX+258+4))

C   FLAGS (12 BITS): F1 = PREEDIT DATA FLAGS, F2 = MDT DATA FLAGS
C                      F3 = IMAGE LINE NUMBER, F4 = SCAN LINE NUMBER
IDFLAG(1) = ISHFT(ICHAR(BUFF(NPIX+262+1)),4) +
+           ISHFT(ICHAR(BUFF(NPIX+262+2)), -4)
IDFLAG(2) = ISHFT(IAND(ICHAR(BUFF(N+262+2)), '0F'Z),8) +
+           ICHAR(BUFF(NPIX+262+3))
IDFLAG(3) = ISHFT(ICHAR(BUFF(NPIX+262+4)),4) +
+           ISHFT(ICHAR(BUFF(NPIX+262+5)), -4)
IDFLAG(4) = ISHFT(IAND(ICHAR(BUFF(NPIX+262+5)), '0F'Z),8) +
+           ICHAR(BUFF(NPIX+262+6))
PRINT '("DataFlags  =" ,4(X,I4))', IDFLAG

RETURN
END

C-----  

C   This Subroutine will print the IR Documentation
C-----  

SUBROUTINE PRIDOC(BUFF)

CHARACTER      BUFF(256)          ! Buffer for record block
INTEGER*2       IRDOC(128)         ! IR Documentation
INTEGER*4       OABLK(6)          ! Orbit/Attitude Block
INTEGER*4       ITMP

C Convert to 16-Bit Integers
DO I=1,128
  IRDOC(I) = ISHFT(IAND(ICHAR(BUFF(2*I-1)), '01'Z),8) +
+             ICHAR(BUFF(2*I))
END DO

C Orbit/Attitude Block are in 32-Bit Integers
DO I=1,6
  ITMP = IRDOC(100+4*I-3)
  ITMP = IRDOC(100+4*I-2)+ISHFT(ITMP,8)
  ITMP = IRDOC(100+4*I-1)+ISHFT(ITMP,8)
  ITMP = IRDOC(100+4*I-0)+ISHFT(ITMP,8)
  OABLK(I) = ITMP
END DO

PRINT '("---- IR Doc ----")'
PRINT '("Retrace    =" ,X,I3)', IRDOC(1)
PRINT '("Spacecraft =" ,X,I3)', IRDOC(2)
C   PRINT '("Unused    =" ,X,I3)', IRDOC(3)
PRINT '("FrameCode  =" ,X,I3)', IRDOC(4)
PRINT '("ChangeCode =" ,X,I3)', IRDOC(5)
PRINT '("StepCode   =" ,X,I3)', IRDOC(6)
PRINT '("LineDelay  =" ,X,I3)', IRDOC(7)

```

```

PRINT '("IRSelect      =" , X, I3)', IRDOC(8)
PRINT '("GrySclStat   =" , X, I3)', IRDOC(9)
PRINT '("XmitMode     =" , X, I3)', IRDOC(10)
PRINT '("ImgLineNo    =" , 4(X, I3))',
+ IAND(ISSHFT(IRDOC(11), -4), '0F'Z), IAND(IRDOC(11), '0F'Z),
+ IAND(ISSHFT(IRDOC(12), -4), '0F'Z), IAND(IRDOC(12), '0F'Z)
PRINT '("ModeCount    =" , I3)', IRDOC(13)
PRINT '("BetaCount    =" , 3(X, I3))', IRDOC(14:16)
PRINT '("GridNoGrid   =" , X, I3)', IRDOC(17)
PRINT '("SyncError    =" , 2(X, I3))', IRDOC(18:19)
PRINT '("BitErrCnt    =" , 2(X, I3))', IRDOC(20:21)
PRINT '("SetupError   =" , X, I3)', IRDOC(22)
PRINT '("CompError    =" , 2(X, I3))', IRDOC(23:24)
PRINT '("ScanCount    =" , 4(X, I3))',
+ IAND(ISSHFT(IRDOC(25), -4), '0F'Z), IAND(IRDOC(25), '0F'Z),
+ IAND(ISSHFT(IRDOC(26), -4), '0F'Z), IAND(IRDOC(26), '0F'Z)
PRINT '("XmitMode     =" , X, I3)', IRDOC(10)
PRINT '("ImgLineNo    =" , 4(X, I3))',
+ IAND(ISSHFT(IRDOC(11), -4), '0F'Z), IAND(IRDOC(11), '0F'Z),
+ IAND(ISSHFT(IRDOC(12), -4), '0F'Z), IAND(IRDOC(12), '0F'Z)
PRINT '("ModeCount    =" , I3)', IRDOC(13)
PRINT '("BetaCount    =" , 3(X, I3))', IRDOC(14:16)
PRINT '("GridNoGrid   =" , X, I3)', IRDOC(17)
PRINT '("SyncError    =" , 2(X, I3))', IRDOC(18:19)
PRINT '("BitErrCnt    =" , 2(X, I3))', IRDOC(20:21)
PRINT '("SetupError   =" , X, I3)', IRDOC(22)
PRINT '("CompError    =" , 2(X, I3))', IRDOC(23:24)
PRINT '("ScanCount    =" , 4(X, I3))',
+ IAND(ISSHFT(IRDOC(25), -4), '0F'Z), IAND(IRDOC(25), '0F'Z),
+ IAND(ISSHFT(IRDOC(26), -4), '0F'Z), IAND(IRDOC(26), '0F'Z)
PRINT '("CorrYear     =" , 4(X, I3))',
+ IAND(ISSHFT(IRDOC(27), -4), '0F'Z), IAND(IRDOC(27), '0F'Z),
+ IAND(ISSHFT(IRDOC(28), -4), '0F'Z), IAND(IRDOC(28), '0F'Z)
PRINT '("CorrDay      =" , 4(X, I3))',
+ IAND(ISSHFT(IRDOC(29), -4), '0F'Z), IAND(IRDOC(29), '0F'Z),
+ IAND(ISSHFT(IRDOC(30), -4), '0F'Z), IAND(IRDOC(30), '0F'Z)
PRINT '("CorrHour     =" , 2(X, I3))',
+ IAND(ISSHFT(IRDOC(31), -4), '0F'Z), IAND(IRDOC(31), '0F'Z)
PRINT '("CorrMin      =" , 2(X, I3))',
+ IAND(ISSHFT(IRDOC(32), -4), '0F'Z), IAND(IRDOC(32), '0F'Z)
PRINT '("CorrSec      =" , 2(X, I3))',
+ IAND(ISSHFT(IRDOC(33), -4), '0F'Z), IAND(IRDOC(33), '0F'Z)
PRINT '("CorrMSec     =" , 2(X, I3))',
+ IAND(ISSHFT(IRDOC(34), -4), '0F'Z), IAND(IRDOC(34), '0F'Z)
PRINT '("BlkEnable    =" , X, I3)', IRDOC(35)
PRINT '("ModeC_Cal   =" , X, I3)', IRDOC(36)
PRINT '("BFSyncLock   =" , X, I3)', IRDOC(37)
PRINT '("LimScnMode   =" , X, I3)', IRDOC(38)
PRINT '("SmpScnMode   =" , X, I3)', IRDOC(39)
PRINT '("VisChCnct   =" , X, I3)', IRDOC(40)
PRINT '("ScanDir      =" , X, I3)', IRDOC(41)
PRINT '("BiPhaseMod   =" , X, I3)', IRDOC(42)
C PRINT '("Unused       =" , X, I3)', IRDOC(43)
PRINT '("PLLError     =" , X, I3)', IRDOC(44)
PRINT '("TestData     =" , X, I3)', IRDOC(45)
PRINT '("DataRandom   =" , X, I3)', IRDOC(46)
PRINT '("SunPulsSel   =" , X, I3)', IRDOC(47)

```

C

```

PRINT '("NessMode      =" ,X,I3)', IRDOC(48)
PRINT '("LimScnCmd    =" ,X,I3)', IRDOC(49)
PRINT '("DigSunPuls   =" ,X,I3)', IRDOC(50)
PRINT '("BitErrLght   =" ,X,I3)', IRDOC(51)
PRINT '("MeanIRDiff   =" ,X,I3)', IRDOC(52)
PRINT '("RMSIRDiff   =" ,X,I3)', IRDOC(53)
PRINT '("CorrectTbl   =" ,X,I6)', ISHFT(IRDOC(54),8)+IRDOC(55)
PRINT '("LfHorizPt   =" ,X,I6)', ISHFT(IRDOC(56),8)+IRDOC(57)
PRINT '("RtHorizPt   =" ,X,I6)', ISHFT(IRDOC(58),8)+IRDOC(59)
PRINT '("EqScanCnt   =" ,X,I6)', ISHFT(IRDOC(60),8)+IRDOC(61)
C PRINT '("Unused       =" ,2(X,I3))', IRDOC(62:79)
PRINT '("TelemCode   =" ,X,I3)', IRDOC(80)
PRINT '("QtrTelem    =" ,18(X,I3))', IRDOC(81:98)
PRINT '("OABLkNum   =" ,X,I3)', IRDOC(99)
PRINT '("QtrNum      =" ,X,I3)', IRDOC(100)
PRINT '("OABlock     =" ,6(X,I10))', OABLK
C PRINT '("Unused       =" ,3(X,I3))', IRDOC(125:127)
PRINT '("LonParChk   =" ,X,I3)', IRDOC(128)
PRINT '("-----")'

```

```

RETURN
END

```

```

C-----
C      This Function returns EBCDIC to ASCII character index
C-----
FUNCTION IEBC(I)

```

```

INTEGER EBCTBL(256)

```

```

DATA EBCTBL /
+ 000,001,002,003,255,009,255,127,255,255,255,011,012,013,014,015, ! 0_
+ 016,017,018,019,255,133,008,255,024,025,255,255,028,029,030,031, ! 1_
+ 255,255,255,255,255,010,023,027,255,255,255,255,005,006,007, ! 2_
+ 255,255,022,255,255,255,004,255,255,255,255,020,021,255,026, ! 3_
+ 032,255,255,255,255,255,255,255,255,162,046,060,040,043,124, ! 4_
+ 038,255,255,255,255,255,255,255,255,033,036,042,041,059,172, ! 5_
+ 045,047,255,255,255,255,255,255,255,255,166,044,037,095,062,063, ! 6_
+ 255,255,255,255,255,255,255,255,096,058,035,064,039,061,034, ! 7_
+ 255,097,098,099,100,101,102,103,104,105,255,255,255,255,255,177, ! 8_
+ 255,106,107,108,109,110,111,112,113,114,255,255,255,255,255,255, ! 9_
+ 255,126,115,116,117,118,119,120,121,122,255,255,255,255,255,255, ! a_
+ 094,255,255,255,255,255,255,255,091,093,255,255,255,255,255, ! b_
+ 123,065,066,067,068,069,070,071,072,073,255,255,255,255,255,255, ! c_
+ 125,074,075,076,077,078,079,080,081,082,255,255,255,255,255,255, ! d_
+ 092,255,083,084,085,086,087,088,089,090,255,255,255,255,255,255, ! e_
+ 048,049,050,051,052,053,054,055,056,057,255,255,255,255,255,159/ ! f_
C          _0  _1  _2  _3  _4  _5  _6  _7  _8  _9  _A  _B  _C  _D  _E  _F

```

```

IEBC = EBCTBL(I+1)

```

```

RETURN
END

```

```

C-----
C ^FUNCTION: I2SWAP
C
C      This function will swap the bytes of a 2-byte word
C-----

INTEGER*2 FUNCTION I2SWAP(BUFF)

CHARACTER          BUFF(2)          ! Input data buffer
CHARACTER          TEMP(2)          ! Output swapped buffer
INTEGER*2           I2BUFF
EQUIVALENCE        (TEMP, I2BUFF)

TEMP(1) = BUFF(2)
TEMP(2) = BUFF(1)
I2SWAP = I2BUFF

RETURN
END

C-----
C ^FUNCTION: I4SWAP
C
C      This function will swap the bytes of a 4-byte word
C-----

INTEGER*4 FUNCTION I4SWAP(BUFF)

CHARACTER          BUFF(4)          ! Input data buffer
CHARACTER          TEMP(4)          ! Output swapped buffer
INTEGER*4           I4BUFF
EQUIVALENCE        (TEMP, I4BUFF)

TEMP(1) = BUFF(4)
TEMP(2) = BUFF(3)
TEMP(3) = BUFF(2)
TEMP(4) = BUFF(1)
I4SWAP = I4BUFF

RETURN
END

```